**Test 1: Valid input file**

* **Purpose/Objective:** To test if the program correctly opens and decodes a compressed file with regular input.
* **Test Configuration:** Run the program with test file compressed.dat

A screenshot of a computer

Description automatically generated

* **Expected Results:** The program should output a file called “compressed\_uncompressed.txt” that contains:

Happy Birthday to You

Happy Birthday to You

* **Actual Results:**

A screenshot of a message

Description automatically generated

* **Pass/Fail:** PASS

**Test 2: Valid input file with punctuation**

* **Purpose/Objective:** To test if the program correctly opens and decodes a compressed file with punctuation.
* **Test Configuration:** Run the program with test file compressed.dat

A screenshot of a computer

Description automatically generated

* **Expected Results:** The program should output a file called “compressed\_uncompressed.txt” that contains:

Happy Birthday to You

Happy Birthday to You

Happy Birthday Dear (name)

Happy Birthday to You.

From good friends and true,

From old friends and new,

May good luck go with you,

And happiness too.

Alternative ending:

How old are you?

How old are you?

How old, How old

How old are you?

* **Actual Results:**

A screenshot of a message

Description automatically generated

* **Pass/Fail:** PASS

**Test 3: Valid input file that starts with punctuation**

* **Purpose/Objective:** To test if the program correctly opens and decodes a compressed file that starts with punctuation.
* **Test Configuration:** Run the program with test file input.txt

A screenshot of a message

Description automatically generated

* **Expected Results:** The program should output a file called “input\_uncompressed.txt” that contains:

..Dear Sally,

Please, please do it-it would please

Mary very, very much. And Mary would

do everything in Mary's power to make

it pay off for you.

-- Thank you very much –

* **Actual Results:**

A screenshot of a message

Description automatically generated

* **Pass/Fail:** PASS

**Test 4: Valid input file that doesn’t have ‘0’ on a line by itself**

* **Purpose/Objective:** To test if the program correctly opens and decodes a compressed file that doesn’t end with ‘0’
* **Test Configuration:** Run the program with test file input.txt

A screenshot of a message

Description automatically generated

* **Expected Results:** The program should output a file called “input\_uncompressed.txt” that contains:

..Dear Sally,

Please, please do it-it would please

Mary very, very much. And Mary would

do everything in Mary's power to make

it pay off for you.

-- Thank you very much –

* **Actual Results:**

A screenshot of a chat

Description automatically generated

* **Pass/Fail:** PASS

**Test 5: Valid input file that the last line is ‘0’ followed by more chars.**

* **Purpose/Objective:** To test if the program correctly opens and decodes a compressed file that has lines (or a last line) that has 0 and other text.
* **Test Configuration:** Run the program with test file input.txt

A screenshot of a message

Description automatically generated

* **Expected Results:** The program should output a file called “input\_uncompressed.txt” that contains:

..Dear Sally,

Please, please do it-it would please

0Mary very, very much. And 0Mary would

do everything in 0Mary's power to make

it pay off for you.

-- Thank you very much --

0 b

* **Actual Results:**

A screenshot of a message

Description automatically generated

* **Pass/Fail:** PASS

**Test 6: Index overflow**

* **Purpose/Objective:** To test if the program gracefully ends if an index with an overflow value is provided.
* **Test Configuration:** Run the program with test file input.txt

A screenshot of a message

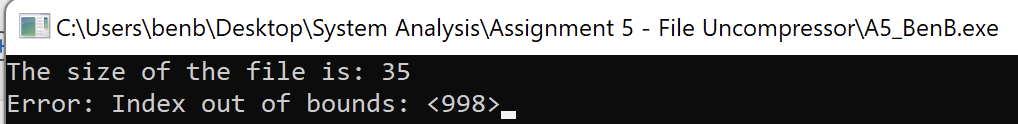
Description automatically generated

* **Expected Results:** The program should output a file called “input\_uncompressed.txt” that contains:

The size of the file is: 35

Error: Index out of bounds: <998>

* **Actual Results:**



* **Pass/Fail:** PASS